

REMARKS

Claims 9-10 and 20-25 have been cancelled as shown on pp. 4-6 of the Reply. Thus, claims 1-7 and 10-19 are currently pending in the subject application and are presently under consideration. Claims 1, 15, and 17 have been amended as shown on p. 2-4 of the Reply.

Favorable reconsideration of the subject patent application is respectfully requested in view of the comments and amendments herein.

I. 3-4-08 Telephonic Interview

Initially, applicants' representative wishes to gratefully acknowledge the Examiner's consideration of the present application *via* a telephonic interview conducted March, 10, 2008. In this regard, Applicant's representative appreciates the Examiner's discussion of the claimed subject matter's limitations, "embedded specification, the specification specified at a source code level" and "a checker . . . that employs the specification" in light of Dollin *et al.* and Cwalina *et al.*, and as they apply to the subject matter of claim 1 (claim 1 of record).

Specifically, it was contended that the program instructions of Dollin *et al.*, from which are derived the type signatures, could also be considered an embedded specification. While applicants' representative disagreed and submitted that, as claimed, it would be understood by one skilled in the art that the embedded specification comprises something more than the program instructions, applicants' representative especially appreciates the discussion of proposed amendments of claim 1.

While no particular form of amendments were agreed upon, it was agreed that further amendments to clarify the subject matter of the embedded would assist the examiner in further consideration of the claims. In this regard, the remaining independent claims were generally discussed in terms of clarifying amendments in relation to the subject matter of claim 1.

Additionally, applicants' representative submitted that the described use of a separate rules assembly by Cwalina *et al.* fails to disclose the receiving executable code with an embedded specification considering that the executable code is checked using the embedded specification as applicants claim. Applicants' representative believes that agreement was reached on this point.

II. Objection of Claims 22 and 23

Claims 22 and 23 are objected to because each claim recites the term “trasmitting” rather than “transmitting.” This rejection is considered moot in light of the cancellation of claims 22 and 23, and thus, the objection should be withdrawn.

IV. Rejection of Claims 1-7 and 11-25 Under 35 U.S.C. § 102(e)

Claims 1-7 and 11-25 stand rejected under 35 U.S.C. § 102(e) as being anticipated by Dollin *et al.*, U.S. Patent No. 7,036,111 (hereinafter “Dollin”). Claims 9-10 and 20-25 have been cancelled, and thus, claims 1, 15, and 17 are the currently pending independent claims. Applicants’ representative respectfully requests withdrawal of this rejection, because Dollin does not expressly or inherently describe each and every limitation of applicants’ claimed invention.

For a prior art reference to anticipate, 35 U.S.C. § 102 requires that “***each and every element*** as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.” *In re Robertson*, 169 F.3d 743, 745, 49 USPQ2d 1949, 1950 (Fed. Cir. 1999) (*quoting Verdegaal Bros., Inc. v. Union Oil Co.*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987)) (emphasis added).

The disclosed subject matter relates to static checking of executable object code, and more particularly, to a system and method employing pre- and/or post- condition(s) specified at a source code level and persisted (*e.g.*, in associated object code and/or a specification repository) facilitating static checking of the object code. The system and method are based, at least in part, upon a framework that employs rules for using an interface to be recorded as declarative specifications in an existing language.

In one aspect, the disclosed subject matter can employ a framework for declaration of pre-condition(s) and/or post-condition specification(s) at a source code level (*e.g.*, embedded within source code). *See, e.g.*, p. 6, ll. 19-21. The pre-condition and/or post-condition specification(s) can include, for example, rule(s) for using an interface, system resource management, order of method call(s) and/or formatting of string parameters. *See, e.g.*, p. 6, ll. 23-24.

The executable code (*e.g.*, object code) with specification(s) can then be provided to a checker. The checker can employ the specification to facilitate static checking of the object file

and provide information if a fault condition (e.g., error(s)) is determined to exist. As a result, programming error(s) that could cause run-time exception(s) and that could escape traditional testing methods (e.g., static source code checking at compile time) can be mitigated.

In a further aspect, once the system has employed the embedded specification, the embedded specification can, optionally, be removed from the executable code, thus, reducing the overall physical storage requirements associated with the executable code. *See, e.g.*, p. 3, ll. 15-18.

In contrast, Dollin relates to code verification systems for type checking compiled code. To that end, Dollin discloses a code verifier that analyzes instructions of a compiled program stored in memory that generates a type signatures based on the instructions. *See e.g.*, Abstract. Dollin further describes that the code verifier analyzes each instruction of the code, after subdividing the code into code blocks, and translates each instruction of the code into a corresponding type signature of the instruction to create type signature blocks. *Id.* However, Dollin fails disclose aspects of applicants' claim invention.

For example, regarding independent claim 1, the claim recites: ***an input component operating on computer hardware that receives an executable object file having an embedded specification***. Regarding independent claim 17, the claim similarly recites: ***receiving executable code with an embedded specification***. Column 3, lines 19-24 and column 5, lines 20-34 are cited for support that Dollin discloses these limitations of applicants' claimed invention. At the indicated paragraphs, Dollin fails to explicitly or inherently disclose these aspects of applicants' claimed invention. For instance, at Column 3, lines 19-24, Dollin describes storing a compiled program in memory. Similarly, at column 5, lines 20-34, Dollin describes analyzing instructions of the compiled program to create type signatures or type signatures blocks based on the instructions stored in memory. Notably, these type signatures are first created once the analyzer analyzes the compiled program instructions. Applicants' representative respectfully submits creating type signatures based on analysis of stored instructions of a compiled program cannot be said to explicitly or inherently disclose applicants' claimed invention.

For example, because the Dollin type signatures do not exist but for the analysis of the Dollin code analyzer, the type signatures cannot be said to be embedded in the instructions of the stored program. It is contended that deriving information (type signatures) from information embedded in the program (the actual program instructions) discloses an "embedded

specification” as applicants’ claim. Applicants’ representative disagrees and submits that the terms “embedded” and “embedded specification” is sufficiently distinct from the term “derived” and the acts of deriving type signatures from program instructions as to patentably distinguish applicants’ claimed invention over Dollin. Furthermore, Dollin is silent on any aspect of embedding the type signatures in the object code after the type signatures are created. Thus, Dollin cannot be said to disclose these aspects of applicants’ claimed invention.

In response, it was contended that the program instructions themselves can be considered an embedded specification. Applicants’ representative disagreed and contended that the program instructions of Dollin cannot be said to disclose the embedded specification as applicants claim. For example, the embedded specification can, optionally, be removed from the executable code, thus, reducing the overall physical storage requirements associated with the executable code. *See, e.g., p. 3, ll. 15-18.* Thus, it is inherent that the specification comprises something more than the program instructions because removing the program instructions would create an inoperative object file.

As a further example, independent claims 1 and 15 similarly recite, ***the specification specified at a source code level***. In one aspect, the disclosed subject matter can employ a framework for declaration of pre-condition(s) and/or post-condition specification(s) at a source code level (*e.g., embedded within source code*). *See, e.g., p. 6, ll. 19-21.* It is contended without citation for support in Dollin that deriving type signatures from program instructions necessarily discloses an embedded specification (in the object code) “specified at a source code level.” As described above, the Dollin type signatures do not exist at the level at which the source code is specified, and they are first created when the code analyzer analyzes the compiled program instructions. *See, e.g., column 5, lines 20-34.* Applicants’ representative respectfully submits that deriving information (the type signatures) from compiled program instructions cannot be said to disclose specifying an embedded specification at a source code level. Thus, Dollin cannot be said to disclose these aspects of applicants’ claimed invention.

For further clarification, applicants’ representative has amended currently pending independent claims to similarly recite ***an embedded specification that is removable***. In addition, applicants’ representative has further clarified that the specification is specified at a source code level ***by embedding the specification within source code of the executable object***. Applicants’ representative respectfully submits that Dollin fails to disclose such further

limitations.

Reconsideration and withdrawal of the rejections of independent claims 1, 15, and 17 (and associated dependent claims 2-7, 11-14, 16, and 18-19) under 35 U.S.C. § 102(e) is respectfully requested in view of the comments above.

V. Rejection of Claims 1-7 and 9-25 Under 35 U.S.C. §102(e)

Claims 1-7 and 9-25 stand rejected under 35 U.S.C. §102(e) as being anticipated by Cwalina *et al.*, U.S. Patent No. 7,150,008 (hereinafter “Cwalina”). Claims 9-10 and 20-25 have been cancelled, and thus, claims 1, 15, and 17 are the currently pending independent claims. Applicants’ representative respectfully requests withdrawal of this rejection, because Cwalina does not expressly or inherently describe each and every limitation of applicants’ claimed invention.

Cwalina relates to binary analysis of software assemblies. To that end, Cwalina describes an analysis engine that receives a rule assembly representing one or more rules and a target assembly representing one or more program elements and one or more program element behaviors, each of the target and rule assemblies comprising metadata and intermediate language instructions. *See e.g.*, Abstract. Cwalina further describes identifying a program element or program element behavior from the target assembly that is to be checked for adherence to a rule from the rule assembly and applying the rule to the program element or program element behavior to check the program element or program element behavior for adherence to the rule. *Id.* However, Cwalina fails disclose aspects of applicants’ claim invention.

For example, regarding independent claim 1, the claim recites: ***an input component operating on computer hardware that receives an executable object file having an embedded specification***. Regarding independent claim 17, the claim similarly recites: ***receiving executable code with an embedded specification***. Steps 301 and 302 in FIG. 3, column 8, lines 20-21, column 8, lines 28-36, and column 10, lines 34-40 are cited for support that Cwalina discloses these limitations of applicants’ claimed invention. At the indicated paragraphs, Cwalina fails to explicitly or inherently disclose these aspects of applicants’ claimed invention.

For instance, at steps 301 and 302 in Figure 3 and column 8, lines 20-21, Cwalina describes receiving a rule assembly and a target assembly, which can each be language independent portable executables, for example, created by compiling source code with an

appropriately configured compiler. *See, e.g.*, col. 8, ll. 6-11. As described in Cwalina, the metadata and intermediate language instructions are created by compiling the source code into the language independent portable executables. *Id.* Metadata describes the types, members (*e.g.*, methods, fields, properties, events), and attributes defined in the source code. *See, e.g.*, col. 8, ll. 11-13. Intermediate language instructions are operation codes that perform operations on portions of metadata (*e.g.*, metadata tokens) when a portable executable is executed. *See, e.g.*, col. 8, ll. 13-16. Thus, a just-in-time compiler can translate language independent portable executables into machine specific executable code that can facilitate the coordination of operation codes and portions of metadata at the time of execution. Accordingly, applicants' representative respectfully submit that receiving portable executables in the form of a target assembly or a rule assembly, each comprising metadata and intermediate language instructions to be compiled by a just-in-time compiler, do not expressly or inherently describe ***receiving executable code with an embedded specification*** as applicants claim.

Similarly, at column 8, lines 28-36 and column 10, lines 34-40, Cwalina describes receiving the rule assembly and an exemplary rules assembly source code portion. Notably, the rule assembly is separate from the target assembly and the rule assembly represents the one or more rules, whereas the target assembly represents one or more program elements and one or more program element behaviors to be checked. *See, e.g.*, Abstract and steps 301 and 302 in Figure 3. Applicants' representative respectfully submits receiving two separate assemblies cannot be said to explicitly or inherently disclose applicants' claimed invention. For example, because the Cwalina rule assembly is separate from the target assembly (*e.g.*, the assembly to be checked), the rule assembly cannot be said to be an embedded specification in the executable object code.

While Cwalina discloses that the rule assembly itself can be executable and can be created by compiling source code, such disclosure does not cure the deficiencies of Cwalina, because as applicants claim in independent claim 1 for example, the executable code having the embedded specification is checked by a ***checker operating on computer hardware that employs the specification to facilitate static checking of the executable object file***. Thus it is understood that the specification is embedded in the object file to be checked. Thus, the Cwalina rules assembly fails to disclose ***receiving executable code with an embedded specification***, as applicants claim.

As a further example, independent claims 1 and 15 similarly recite, ***the specification specified at a source code level***. In one aspect, the disclosed subject matter can employ a framework for declaration of pre-condition(s) and/or post-condition specification(s) at a source code level (*e.g.*, embedded within source code). *See, e.g.*, p. 6, ll. 19-21. Column 10, lines 34-40 are cited for support in Cwalina that creating rules in a source code file necessarily discloses an embedded specification (in the object code) “specified at a source code level.” As described above, the Cwalina rules assembly is separate from the target assembly (that which is to be checked). *See, e.g.*, Abstract and steps 301 and 302 in Figure 3. Applicants’ representative respectfully submits that using a separate executable rules file to check a further disparate executable target file cannot be said to disclose these aspects of applicants’ claimed invention.

For further clarification, applicants’ representative has amended currently pending independent claims to similarly recite ***an embedded specification that is removable***. In addition, applicants’ representative has further clarified that the specification is specified at a source code level ***by embedding the specification within source code of the executable object***. Applicants’ representative respectfully submits that Cwalina fails to disclose such further limitations.

Reconsideration and withdrawal of the rejections of independent claims 1, 15, and 17 (and associated dependent claims 2-7, 11-14, 16, and 18-19) under 35 U.S.C. § 102(e) is respectfully requested in view of the comments above.

VI. Rejection of Claims 9 and 10 Under 35 U.S.C. §103(a)

Claims 9 and 10 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Dollin, as applied to claim 1 above, in view of Alaluf, U.S. Pub. No. 2004/0230958 to Alaluf. This rejection is considered moot in light of the cancellation of claims 9 and 10, and thus, the rejection should be withdrawn.

CONCLUSION

The present application is believed to be in condition for allowance in view of the above comments and amendments. A prompt action to such end is earnestly solicited.

In the event any fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063 [MSFTP464US].

Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicants' undersigned representative at the telephone number below.

Respectfully submitted,

AMIN, TUROCY & CALVIN, LLP

/Himanshu S. Amin/

Himanshu S. Amin

Reg. No. 40,894

AMIN, TUROCY & CALVIN, LLP
24TH Floor, National City Center
1900 E. 9TH Street
Cleveland, Ohio 44114
Telephone (216) 696-8730
Facsimile (216) 696-8731